



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Kerja praktek magang dilaksanakan pada *department* atau bagian *technology* di perusahaan PT GLOBAL TIKET NETWORK tepatnya di divisi *tech platform*, tim *utilities*. Pengerjaan koordinasi dilakukan sesuai arahan dari supervisor atau *tech lead*. Dalam pengerjaan proyek API dashboard *whatsapp bulk* terdapat 3 orang rekan kerja, berupa 2 orang *junior backend engineer* yang baru masuk perusahaan. Dan 2 *middle backend engineer* yang sudah memiliki pengalaman kerja.

Table 1 Tabel pembagian tugas

Nama	Tugas
Izdihar Farahdina	Mengerjakan API untuk whatsapp template dashboard yang nantinya template ini akan digunakan untuk <i>bulking</i> .
Didit Suryadi	Mengerjakan API untuk whatsapp template dashboard yang nantinya template ini akan digunakan untuk <i>bulking</i> .
Fransiska Nadapdap	Mengerjakan API untuk dashboard dan whatsapp bulk.
Justin Susanto	Mengerjakan API untuk dashboard dan whatsapp bulk.

#### 3.2 Tugas yang Dilaksanakan

Dalam proses kerja magang yang dilakukan pada proyek *Dashboard Whatsapp Bulk*, proses dibagi kembali menjadi beberapa *subtask*. *Subtask* yang

dilakukan seperti, menyediakan *API* untuk mendapatkan *whatsappbulk status*, *submit whatsappbulk request*, *service* untuk melakukan penyimpanan file bulk, mengunduh *whatsappbulk*, dan yang terakhir mendapat data yang telah diolah dalam bentuk *paginated* untuk ditampilkan pada *dashboard whatsappbulk*.

Proses pengerjaan proyek dilakukan tahap sebagai berikut:

1. Melakukan pemahaman terhadap kebutuh sistem, lalu mendiskusikan Bersama tim objek apa saja yang dibutuhkan dalam sistem *dashboard* ini, sebelumnya sudah dilakukan grooming untuk fitur *whatsapp bulk* tentang spesifikasi sistem.
2. *Coding*, melakukan penulisan program menggunakan bahasa Java sesuai dengan spesifikasi yang dibutuhkan sistem.
3. *Testing dan debugging*, Setelah melakukan *Coding*, *testing* dilakukan di-*localhost* terlebih dahulu untuk mengurangi kesalahan fatal yang terjadi pada program.
4. *Deployment* ke server yang didekasikan untuk pengujian oleh *Software Quality Assurance* (SQA).
5. *Debugging*, jika ditemukan *issue* yang ditemukan oleh *Software Quality Assurance* (SQA), akan kembali lagi ke proses coding.

### **3.3 Uraian Pelaksanaan Kerja Magang**

Pelaksanaan kerja magang dapat diuraikan sebagai berikut.

Table 2 Tabel kegiatan yang dilakukan saat kerja magang

Minggu	Kegiatan
1.	Unit Test.
2.	Unit Test.
3.	Membantu melakukan <i>re-write repository</i> sms + UnitTest.
4.	Membantu melakukan <i>re-write repository</i> sms + UnitTest.
5.	Menambahkan <i>endpoint + service</i> pada header footer dashboard.
6.	Menambahkan <i>endpoint + service</i> pada header footer dashboard.
7.	Menambahkan <i>endpoint + service</i> pada dashboard untuk fitur <i>whatsappbulk</i> .
8.	Menambahkan <i>endpoint + service</i> pada dashboard untuk fitur <i>whatsappbulk</i> .
9.	Menambahkan <i>endpoint + service</i> pada dashboard untuk fitur <i>whatsappbulk</i> .
10.	Menambahkan <i>endpoint + service</i> pada dashboard General Config
11.	Menambahkan <i>endpoint + service</i> pada dashboard General Config

Pelaksanaan kerja magang di minggu pertama, diawali dengan hari pertama ada proses *onboarding* di perusahaan. *On-boarding* merupakan acara untuk penerimaan karyawan baru, pada *On-boarding* karyawan dikenalkan tentang perusahaan. Hari kedua di minggu pertama perkenalan dengan rekan rekan kerja yang dibimbing oleh *tech lead*, dan dikenalkan bagaimana proses penggunaan *framework scrum* di perusahaan.

Pada minggu pertama dan kedua, pembimbing memberikan tugas untuk melakukan unittest, karena unittest membutuhkan pembacaan *code* yang cukup intensif, sekaligus juga mempelajari bagaimana standar dan struktur *code* di

perusahaan dan mempelajari *framework spring boot* dan menggunakan database *mongoDB*.

Pada minggu ketiga dan keempat, pembimbing lapangan memberikan tugas untuk melakukan *rewrite* untuk *repository sms* sekaligus, belajar untuk memahami standar *code* secara langsung dan bagaimana penulisan *code* yang baik dan benar.

Pada minggu kelima dan keenam, pembimbing lapangan memberikan tugas untuk menambahkan *service* dan *endpoint* untuk fitur *header* dan *footer* pada *dashboard home*.

Pada minggu ke ketujuh, kedelapan dan kesembilan , pembimbing lapangan memberikan tugas pada proyek *whatsapp bulk*. Penambahan *endpoint* dan *service* pada *whatsappbulk status*, *submit whatsappbulk request*, *service* untuk melakukan penyimpanan file bulk, mengunduh *whatsappbulk*, dan yang terakhir mendapat data yang telah diolah dalam bentuk paginated untuk ditampilkan pada *dashboard whatsappbulk*. Jika pada minggu kesepuluh saat dilakukan *testing* oleh *Software Quality Assurance (SQA)* dan jika ada *bug* yang ditemukan saat di *testing* di *server testing*, maka *bug* akan langsung diperbaiki.

Pada minggu ke sepuluh, dan kesebelas. Pembimbing lapangan memberikan tugas pada proyek *dashboard General Config*. *EndPoint* dan *Service* pada *General Config* cukup luas. Pembimbing memberikan tugas untuk mengerjakan *EndPoint* dan *Service* pada *General Config* bagian *Currency*.

### 3.3.1 Proses Pelaksanaan

Proses pelaksanaan dari kerja magang yang dilakukan adalah fitur endpoint dan *service* untuk API dashboard *whatsapp bulk* dan juga. *Endpoint* yang diutambahkan berupa, *whatsappbulk status*, *submit whatsappbulk request*, *service* untuk melakukan penyimpanan file bulk, mengunduh *whatsappbulk*, dan *pagination* untuk dashboard whatsapp bulk, menggunakan spesifikasi perangkat lunak sebagai berikut.

1. Sistem Basis Data: MongoDB
2. Sistem manajemen paket atau *Dependency Manager* :  
Apache Maven
3. Perangkat lunak manajemen proyek agile: Atlassian Jira
4. Perangkat lunak pengelola basis data: Robo 3T
5. *Framework*: Spring Boot
6. *Integrated Development Environment* (IDE): IntelliJ IDEA 2019.2 Ultimate (Spring Boot tidak dapat berjalan pada Community version)
7. *Software development platform*: Github
8. *API docs* atau dokumen aplikasi interface: Swagger UI

Sedangkan perangkat keras yang digunakan adalah notebook Dell Inspiron 13 5000 Series dengan spesifikasi sebagai berikut.

1. Prosessor : Intel Core i7 6500U

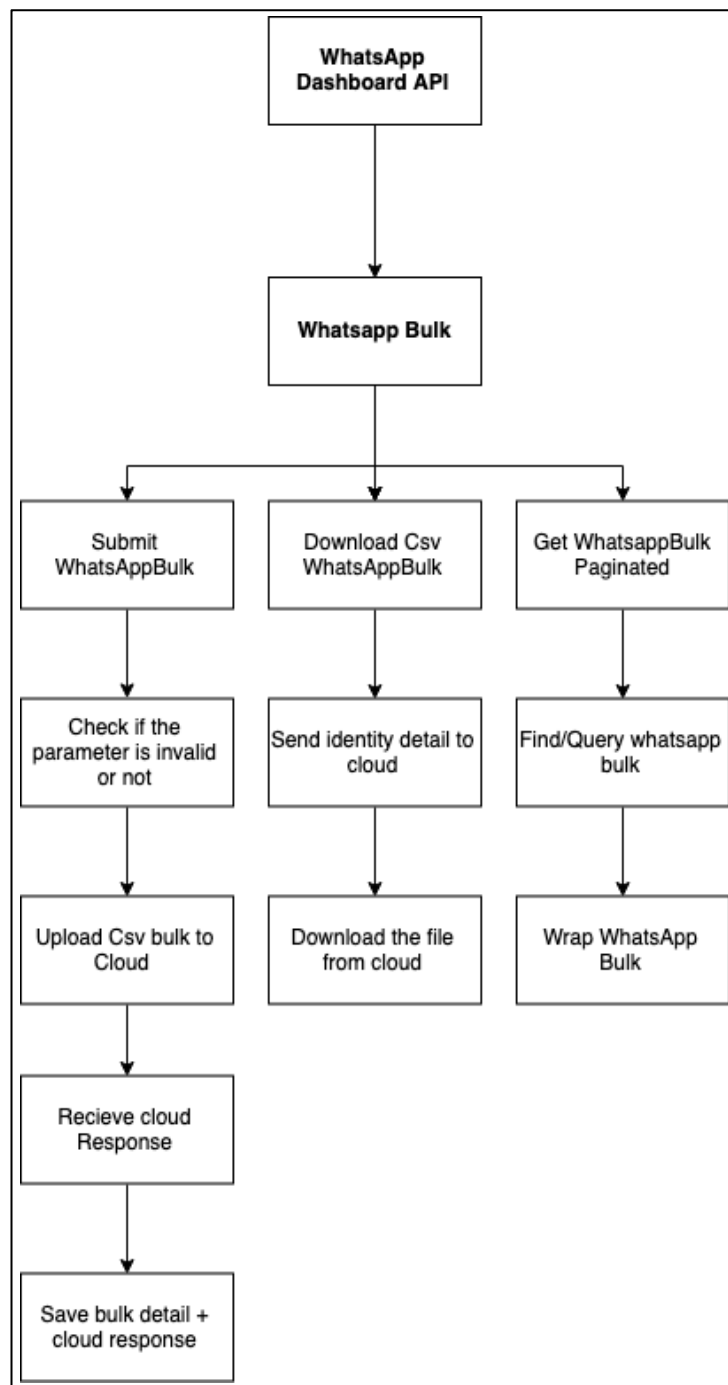
2. Sistem Operasi : Ubuntu Linux (64-bit)
3. Memori : 8 GB LPDDR3
4. Drive : 256 GB

Dengan requirement sebagai berikut:

1. *Whatsappbulk*
  - a. Admin dapat mengirimkan pesan ke banyak pengguna berdasarkan template yang ada.
  - b. Admin dapat melakukan kontrol terhadap konten yang dikirim, termasuk siapa yang membuat konten.
  - c. Diperlukan adanya konfirmasi setiap kali mengirim pesan *bulk*.
2. *General Config*
  - a. Admin dapat melakukan penambahan, perubahan, dan penghapusan data standarisasi *area*, mata uang, dan bahasa.
  - b. Adanya akses terhadap API agar data pada *general config* dapat diakses oleh *microservice* lainnya.

Dengan perancangan sebagai berikut.

## A.1 Rancangan API *Dashboard WhatsApp*



Gambar 3.1 Sitemap API Whatsapp Dashboard

*WhatsApp* API memiliki 9 servis yang dibagi menjadi dua bagian, yaitu *WhatsappTemplate* dan *WhatsappBulk*.

### 1. *WhatsApp Bulk*



a. *Submit WhatsAppBulk*

Servis untuk men-*submit whatsapp bulk* akan mengunggah file CSV yang digunakan untuk melakukan *bulking*. File diunggah ke *cloud service*, balikan yang didapat dari *cloud service* akan disimpan ke dalam *database*.

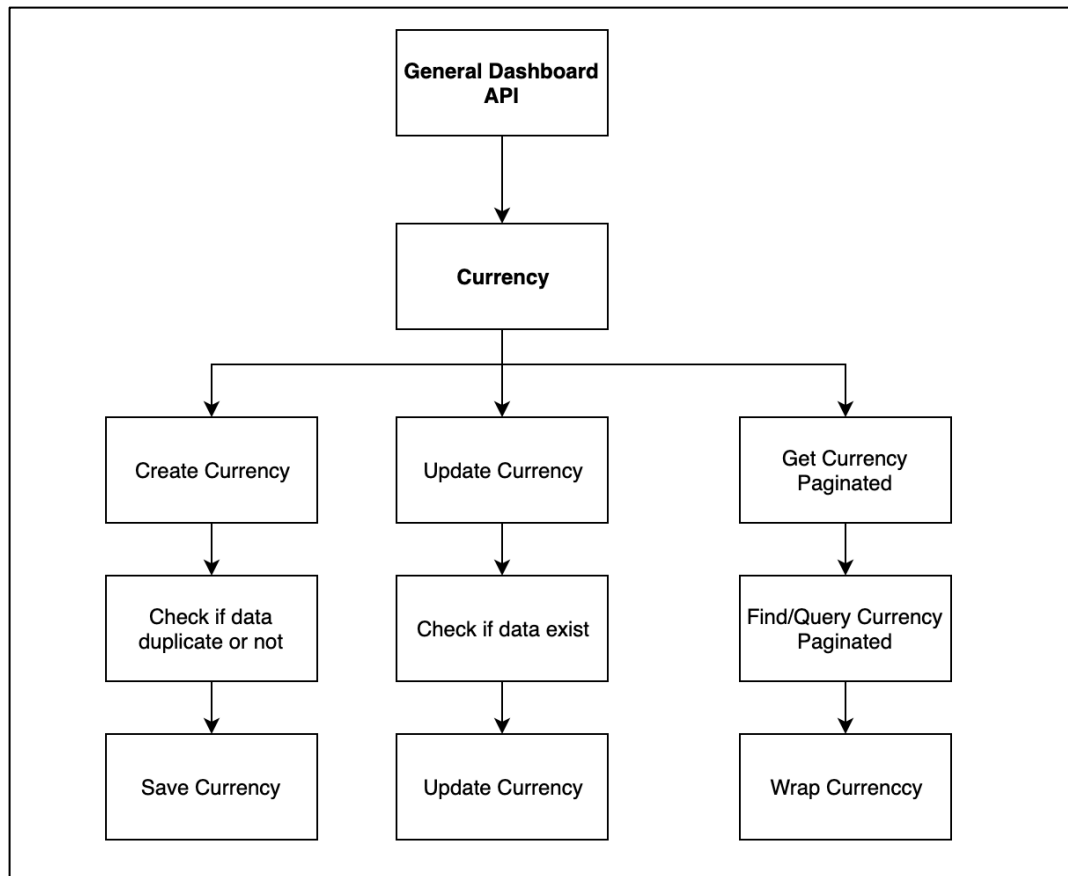
b. *Download Csv WhatsAppBulk*

Servis untuk mengunduh CSV yang di unduh dari *cloud*.

c. *Get WhatsAppBulk Paginated*

Servis untuk mengolah seluruh data *whatsapp bulk* untuk ditampilkan pada *dashboard*, data yang ditampilkan telah di *wrap* dalam bentuk *pagination*. Data yang sudah berbentuk *pagination* ini mudah diolah untuk ditampilkan dalam bentuk halaman.

Dari keseluruhan *project*, karena tidak bisa dibuka keseluruhan datanya dan *Object* pada *database* tidak dapat ditampilkan. *Service* dan *controller* yang ditampilkan hanya sebagian kecil.



Gambar 3. 2 Sitemap API General Config

## A.2 Rancangan API *General Config*

### 1. *General Config*

Fitur general config terdiri dari *area*, *currency*, dan *language*. Yang ditugaskan saat pekerjaan magang beberapa fitur sebagai berikut:

#### a. *Create Currency*

Servis untuk membuat *currency*, yang digunakan untuk generalisasi sistem. Sebelum sukses dalam melakukan *Create Currency* akan diperiksa terlebih dahulu pada *database*, apakah data sebelumnya sudah ada atau belum agar tidak terjadi duplikasi data.

b. *Update Currency*

Servis untuk lakukan *update* pada suatu *currency*, Data yang akan di *replace* juga di cek kembali agar tidak menciptakan data duplikat.

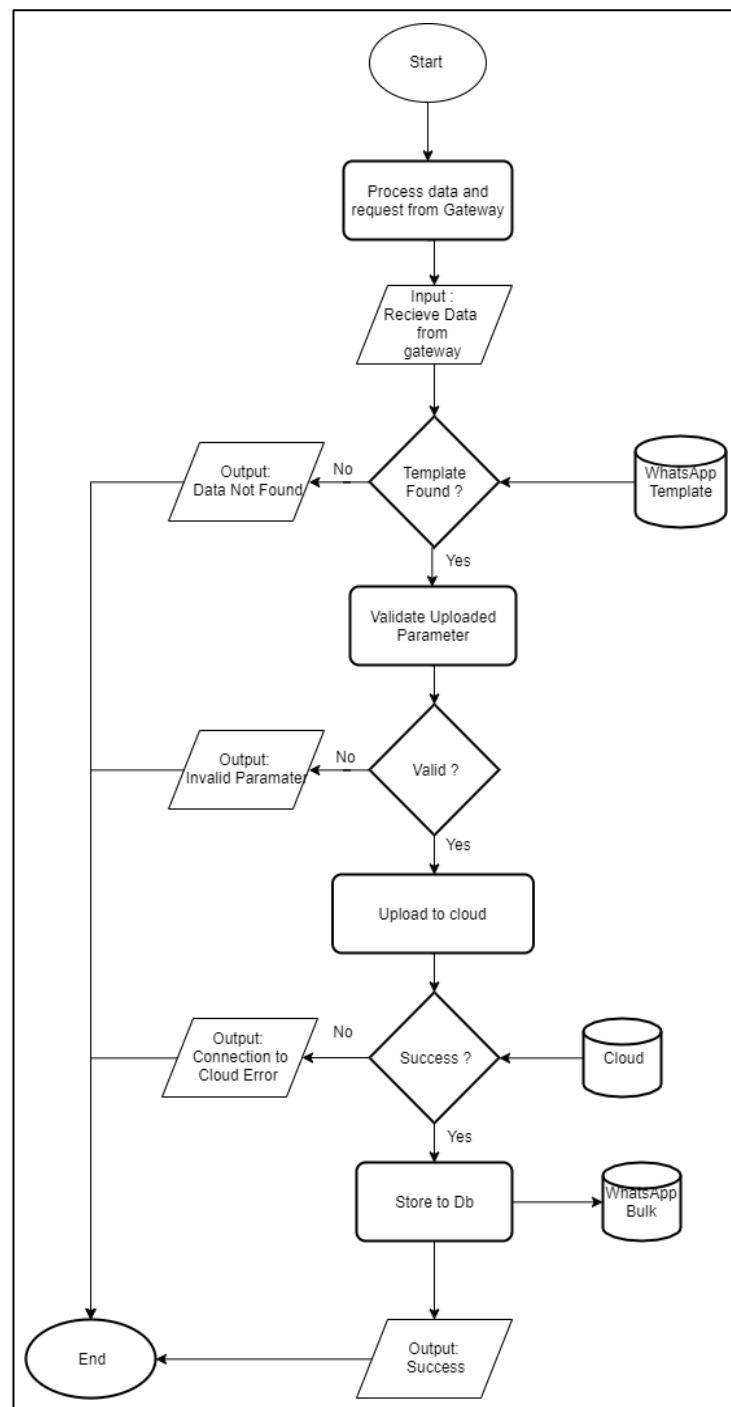
c. *Get Currency Paginated*

Servis untuk mengolah seluruh data yang pada pada *database General Config* untuk ditampilkan pada *dashboard*, data yang ditampilkan telah di *wrap* dalam bentuk *pagination*. Data yang sudah berbentuk *pagination* ini mudah diolah untuk ditampilkan dalam bentuk halaman.

## **A.2 Flowchart**

*Flowchart* pada API *whatsappbulk* dibagi menjadi 3, yaitu *Submit WhatsAppBulk*, *Download CSV WhatsappBulk*, *Get WhatsappBulk Paginated*. Pada API *General Config*, flowchart juga dibagi menjadi 3 yaitu, *Create Currency*, *Update Currency*, dan *Get Currency Paginated*.

### A.2.1.1 Flowchart Submit WhatsApp Bulk

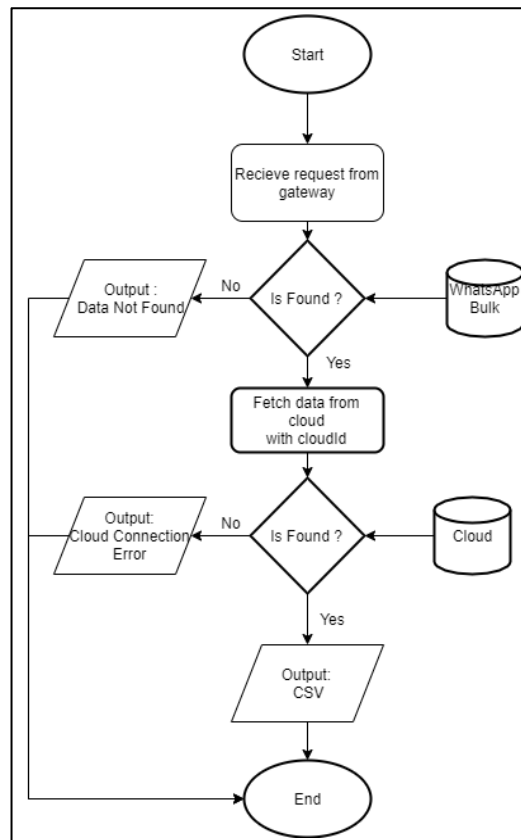


Gambar 3. 3 Flowchart submit WhatsappBulk

Jika proses sukses akan mengoutput *success*. Data yang diupload ke *cloud* telah di *encode* terlebih dahulu. Pertama jika ada *request* ke *endpoint Submit*

*whatsappbulk*, data akan diterima dari *gateway*. Selanjutnya API akan mencari data *Template* pada *database*, jika *template* tidak ditemukan, maka akan memberikan *output* data yang error, dengan *response code* 200, dan message error *Data Not Found*. Jika *template* ditemukan, maka *service whatsappbulk* selanjutnya akan melakukan validasi, terhadap parameter yang dikirim di file CSV. Jika parameter yang dikirim file csv, tidak sama dengan *template* maka sistem akan memberi *response error*, dengan *response code* 200, dan message error *Invalid Parameter*. Parameter yang benar akan dikirim ke cloud dan jika proses pengunggahan file csv ke cloud berhasil, id file yang diberikan *cloud* akan disimpan ke dalam *database whatsapp*. Ketika file yang akan di unggah ke *cloud* mengalami kegagalan, maka API akan memberi *response Connection to cloud error*.

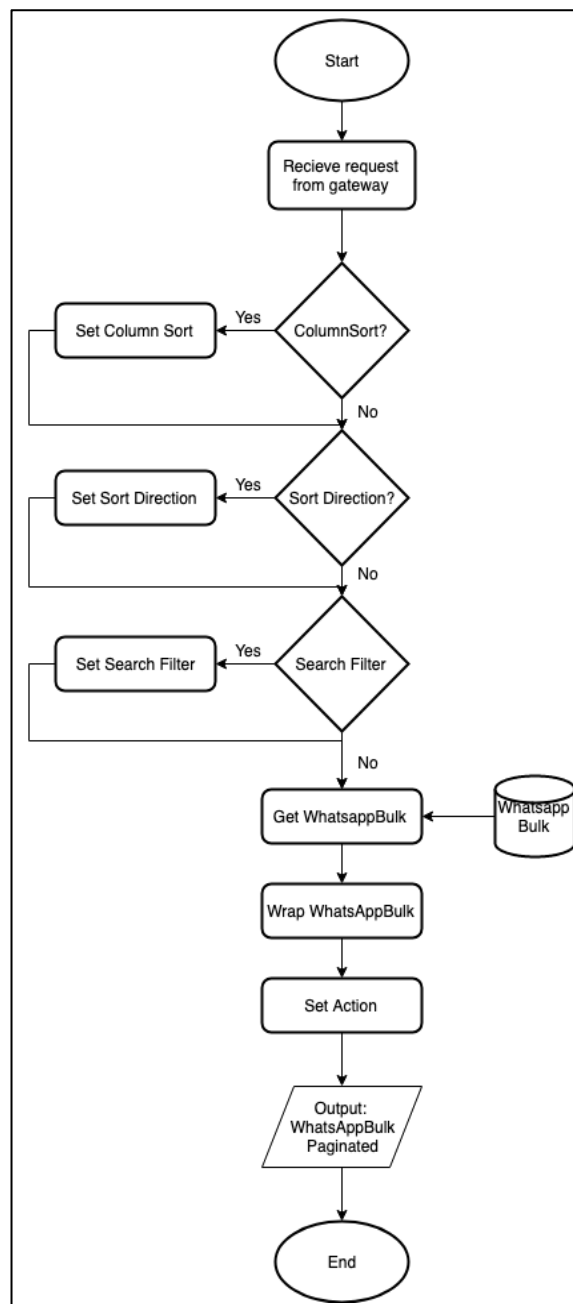
#### A.2.1.2 Flowchart Download Csv WhatsAppBulk



Gambar 3. 4 Flowchart Download Csv WhatsappBulk

Service pada endpoint pada csv whatsapp bulk, jika sukses akan menghasilkan output file csv. Data yang diterima merupakan data yang telah di encode. Pertama request pada endpoint Download CSV whatsappbulk diterima, API Download CSV whatsappbulk, akan melakukan pengecekan terhadap database, dari id bulk yang ingin di download. Jika id bulk ditemukan maka Service akan melakukan pengunduhan file dari cloud dimana file disimpan. Jika tidak berhasil maka sistem akan mengembalikan error, Data Not Found. Jika file juga tidak ditemukan di cloud, maka sistem akan mengembalikan error yang sama, yaitu Data Not Found.

### A 2.1.3 Flowchart WhatsAppBulk Paginated

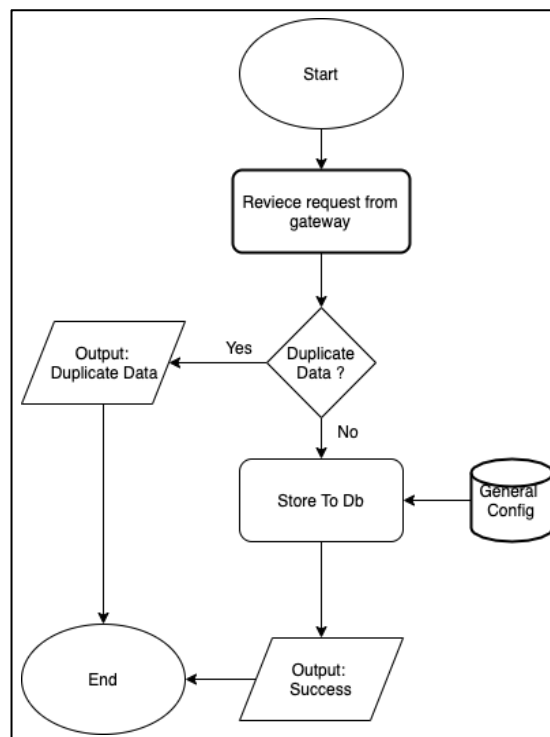


Gambar 3. 5 Flowchart WhatsappBulk Paginated

Service pada *endpoint WhatsAppBulk Paginated*. Pada *endpoint* gambar diatas tidak ada handler, untuk data *whatsappbulk* yang tidak ditemukan. Karena jika data tidak ditemukan data akan mengembalikan data kosong. Data yang akan ditampilkan pada dashboard. Jika data kosong *frontend* akan membuat *handler*.

*Handler* yang dibuat *frontend* untuk memberi informasi tidak ada data pada dashboard. *Column Sort* jika di *set*, menandakan kolom apa yang akan di sort. *Sort Direction* jika di *set*, akan memberikan konfigurasi seperti *Whatsappbulk* akan di sort dengan arah *Ascending* atau *Descending*. *Search Filter* jika di *set*, akan mencari *Whatsappbulk* berdasarkan *template code* atau nama *whatsappbulk*. Secara *default sort* akan dilakukan secara *Descending*. *WhatsAppBulk* di *wrap* menggunakan *PageWrapper* yang merupakan library yang dibuat oleh perusahaan, source code bersifat rahasia.

#### A.2.2.1 Flowchart Create Currency

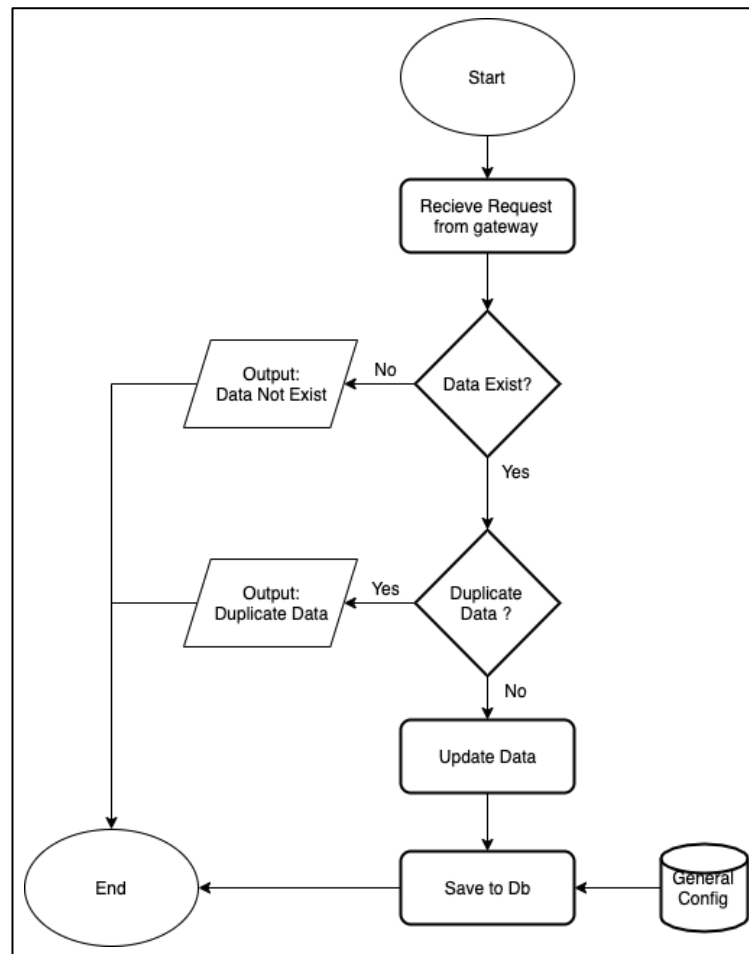


Gambar 3. 6 Flowchart Create Currency

Endpoint untuk melakukan proses *create* pada *currency* yang akan digunakan untuk generalisasi sistem. Pertama *request* ke *endpoint Create Currency*, data akan di lakukan proses pengecekan. Jika ada duplikasi data maka sistem akan mengeluarkan *response code 200*, dengan message *Duplicate data*.



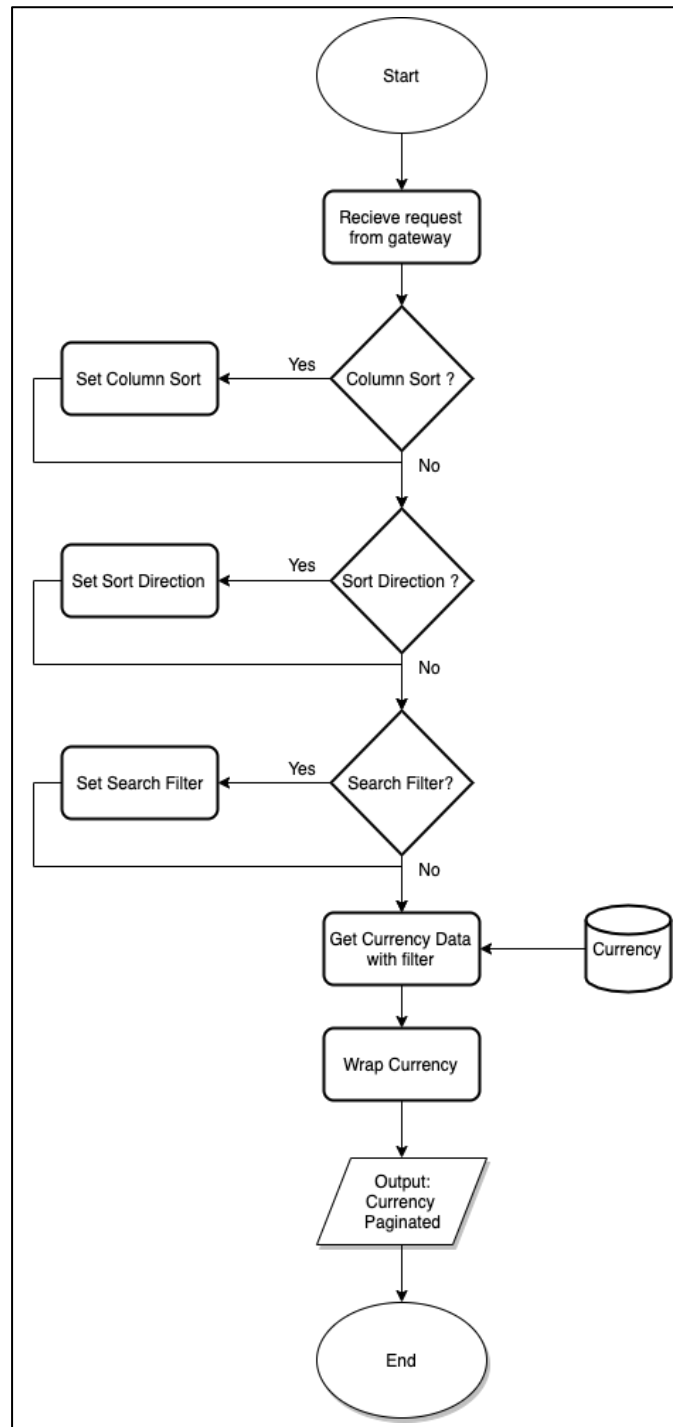
### A.2.2.2 Flowchart Update Currency



Gambar 3. 7 Flowchart Update Currency

Endpoint untuk melakukan *update currency* digunakan untuk memodifikasi atau *edit currency* yang sudah ada, jika ada perubahan data. *Endpoint* untuk melakukan *update currency* akan mengecek jika data ada, dan juga hasil yang di edit akan menghasilkan duplikasi atau tidak. Jika ada duplikasi data dan tidak menemukan data yang akan di *update* maka sistem akan memberi *output*, *Data Not Exist* atau *Duplicate data*. Dengan *response code* 200.

### A.2.2.3 Flowchart Get Currency Paginated

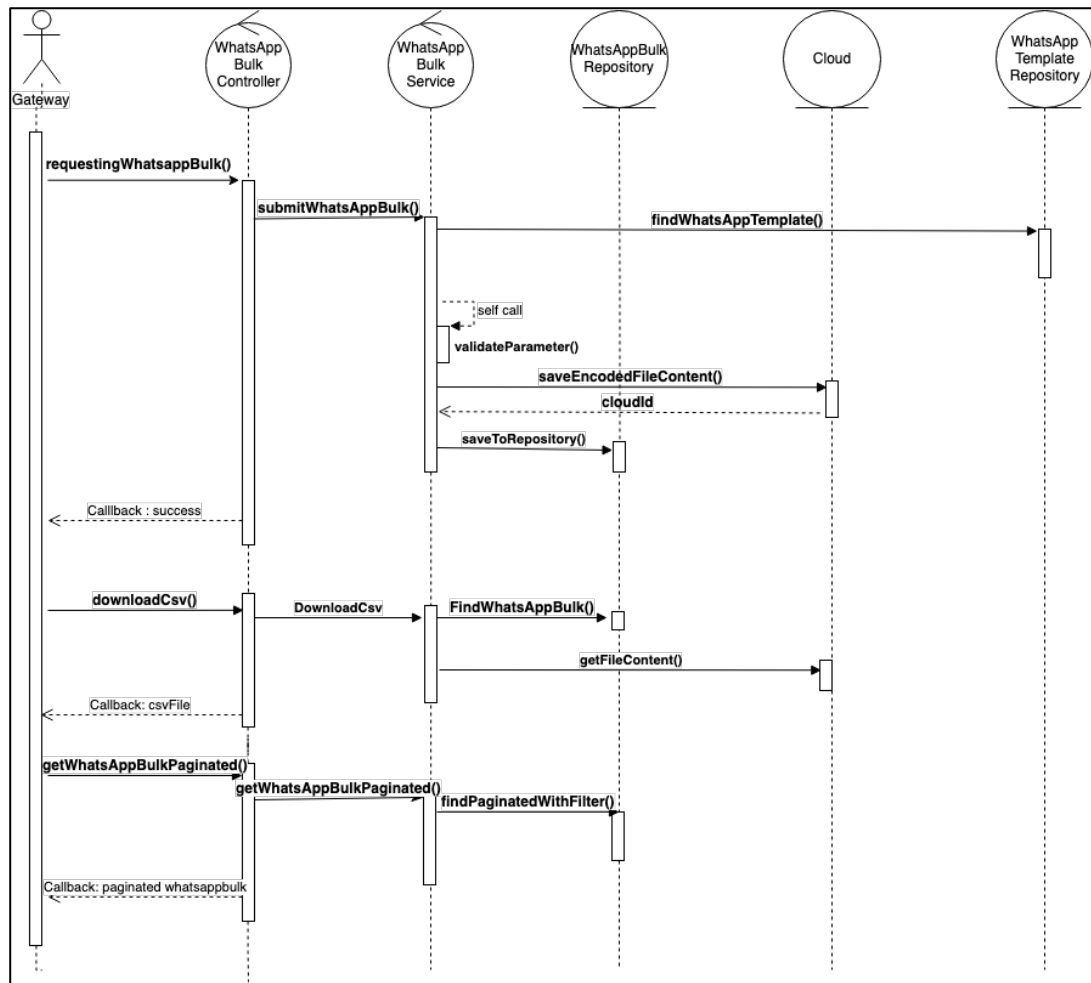


Gambar 3. 8 Flowchart Get Currency Paginated

Service pada *endpoint Currency Paginated*. Pada *endpoint* gambar diatas tidak ada *handler*, untuk data *currency* yang tidak ditemukan. Karena jika data tidak ditemukan data akan mengembalikan data kosong. Data yang akan

ditampilkan pada dashboard. Jika data kosong *front end* akan membuat *handler*. *Handler* yang dibuat *frontend* untuk memberi informasi tidak ada data pada *dashboard*. *Column Sort* jika di *set*, menandakan kolom apa yang akan di sort. *Sort Direction* jika di *set*, akan memberikan konfigurasi seperti *Whatsappbulk* akan di sort dengan arah *Ascending* atau *Descending*. *Search Filter* jika di *set*, akan mencari *Whatsappbulk* berdasarkan *template code* atau nama *whatsappbulk*. *Currency* akan di-wrap menggunakan *PageWrapper* yang merupakan *library* yang dibuat oleh perusahaan, source code bersifat rahasia.

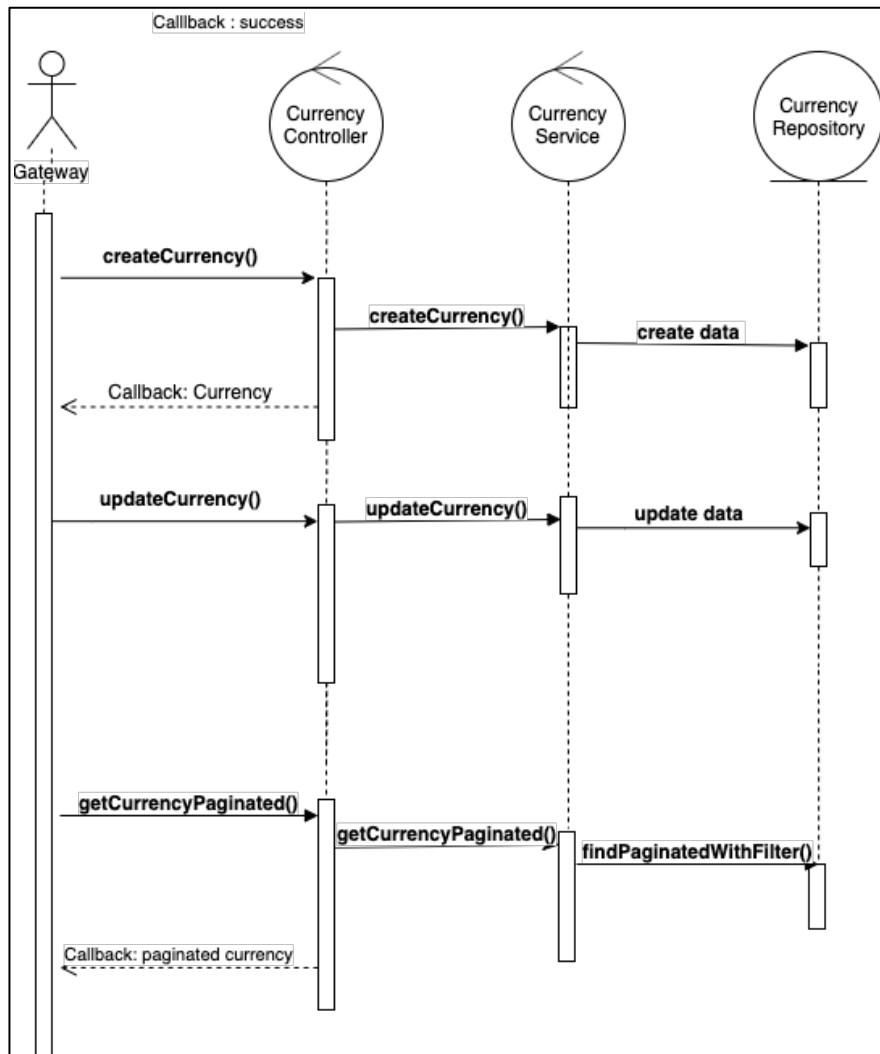
### A.3.1 Sequence Diagram WhatsAppBulk



Gambar 3. 9 Sequence diagram *WhatsappBulk*

*Sequence diagram*, di atas menunjukan bagian sistem *WhatsappBulk* yang dikerjakan. *Gateway* akan melakukan *request* ke *controller*, yang nantinya *controller* akan memberi *response*. *Controller* akan memanggil ke *service*. *Service* akan melakukan *logic* dan memanggil data dari *database* dengan memanggil implementasi dari *Repository*.

### A.3.2 Sequence Diagram General Config



Gambar 3. 10 Sequence diagram General Config

Sequence diagram, diatas menunjukan bagian sistem General Config yang dikerjakan. Gateway akan melakukan request ke controller, yang nantinya controller akan memberi response. Controller akan memanggil ke service. Service akan melakukan logic dan memanggil data dari database dengan memanggil implementasi dari Repository.

## A. Hasil

Karena *Object* dan *code* bersifat rahasia, maka hasil yang akan dilampirkan hanya *response* dari *endpoint* sebagai berikut.

### B.1 Hasil Endpoint *WhatsAppBulk Paginated*

```
},
"pageable": {
  "sort": {
    "sorted": true,
    "unsorted": false,
    "empty": false
  },
  "offset": 0,
  "pageSize": 10,
  "pageNumber": 0,
  "unpaged": false,
  "paged": true
},
"last": false,
"totalPages": 5,
"totalElements": 45,
"size": 10,
"number": 0,
"numberOfElements": 10,
"sort": {
  "sorted": true,
  "unsorted": false,
  "empty": false
},
},
},
```



Gambar 3. 11 Hasil *WhatsApp Bulk Paginated*

### B.2 Hasil Endpoint *Create Currency*

```
Response body
{
  "code": "SUCCESS",
  "message": "SUCCESS",
  "errors": null,
  "data": {
    "id": "5dc1796cef59e8f0500300f3",
    "code": "IDN",
    "name": "Rupiah",
    "specialSymbol": "Rp",
    "specialSymbolPlacing": "BEFORE_NUMBER",
    "areaList": []
  },
  "serverTime": 1572960620107
}
```



Gambar 3. 12 Hasil *Endpoint Create Currency*

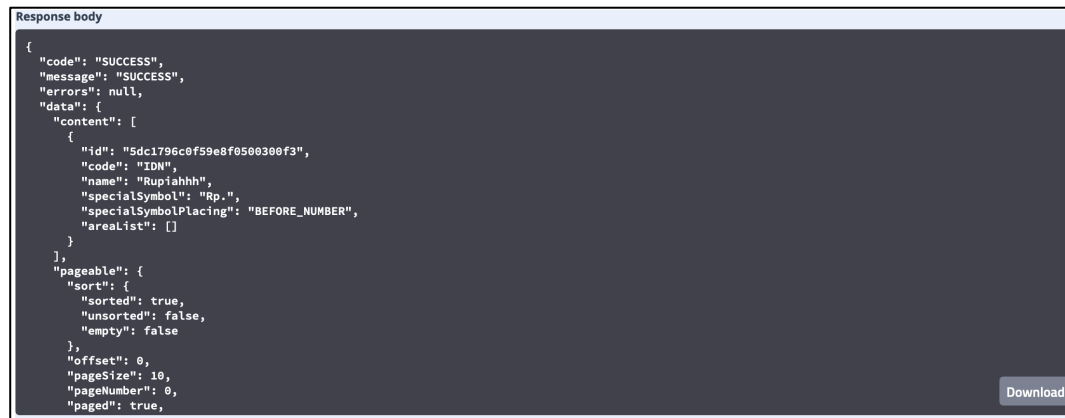
### B.3 Hasil *Endpoint Update Currency*

```
Response body
{
  "code": "SUCCESS",
  "message": "SUCCESS",
  "errors": null,
  "data": {
    "id": "5dc1796cef59e8f0500300f3",
    "code": "IDN",
    "name": "Rupiahhh",
    "specialSymbol": "Rp.",
    "specialSymbolPlacing": "BEFORE_NUMBER",
    "areaList": []
  },
  "serverTime": 1572961011997
}
```



Gambar 3. 13 Hasil *Endpoint Update Currency*

## B.4 Hasil *Endpoint Get Currency Paginated*



```
Response body
{
  "code": "SUCCESS",
  "message": "SUCCESS",
  "errors": null,
  "data": {
    "content": [
      {
        "id": "5dc1796c8f59e8f0500300f3",
        "code": "IDM",
        "name": "Rupiah",
        "specialSymbol": "Rp.",
        "specialSymbolPlacing": "BEFORE_NUMBER",
        "areaList": []
      }
    ],
    "pageable": {
      "sort": {
        "sorted": true,
        "unsorted": false,
        "empty": false
      },
      "offset": 0,
      "pageSize": 10,
      "pageNumber": 0,
      "paged": true
    }
  }
}
```

Gambar 3. 14 Hasil *Endpoint Get Currency Paginated*

### 3.3.2 Kendala yang Ditemukan

Kendala yang saat pengerjaan *dashboard whatsapp bulk* berupa:

1. Belum adanya penambahan logging aktivitas yang dilakukan user.
2. Kesalahan yang sebelumnya sudah ada pada *template whatsapp* saat *development*, membuat *effort* tambahan untuk memperbaiki *template* yang salah.

### 3.3.3 Solusi atas kendala yang Ditemukan

Solusi dari masalah yang ditemukan adalah:

1. Menambahkan *logging* pada *endpoint* tiap aktivitas yang melakukan perubahan/manipulasi data.

2. Memperbaiki template yang sudah ada melalui migrasi data.